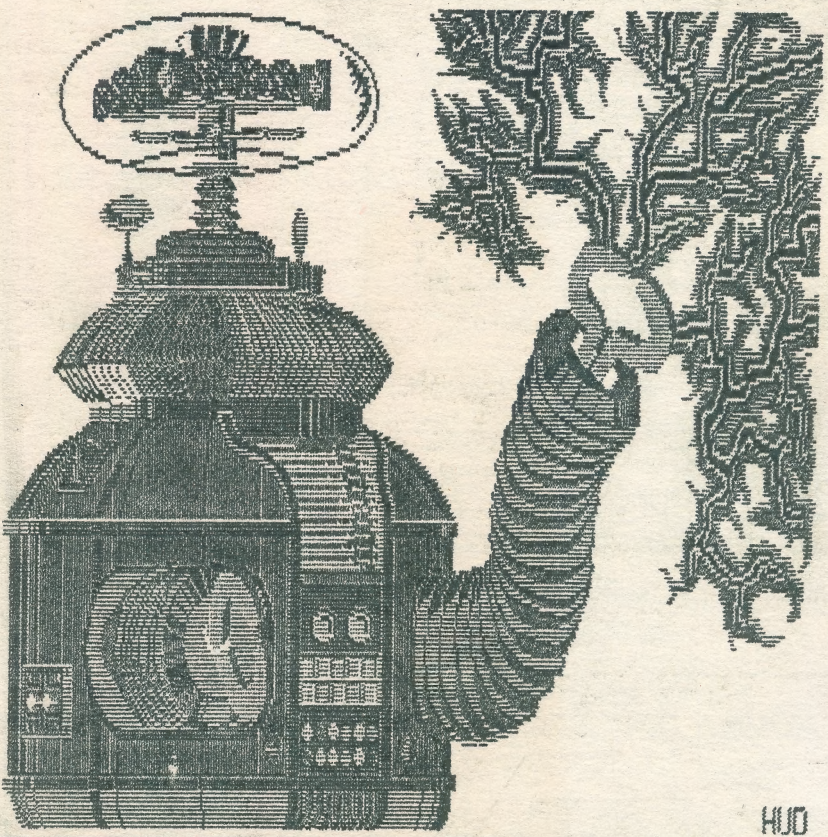
ASTUN ATARI ST USER'S NETWORK



HUD

ASTUN
6157 South 700 West Murray, Utah 84123

Bulk Rate
U.S. Postage
PAID
Midvale Utah
Permit No. 628

AUG-87 C

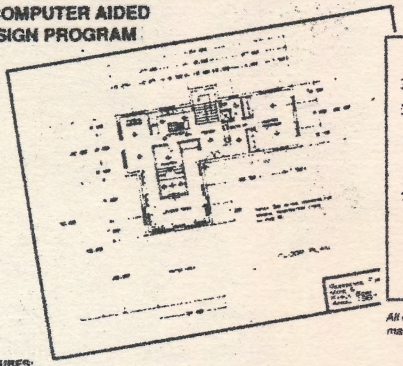
PAT JOHNSON
2818 W. 5825 S.
ROY, UTAH 84067

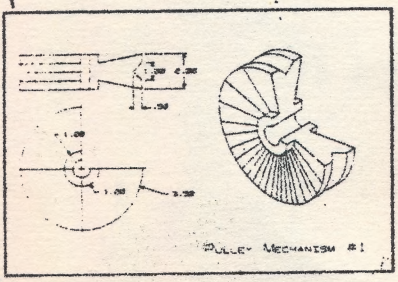
ADDRESS CORRECTION REQUESTED

ATHENA II

A professional, full color, Two-Dimensional Computer Aided Design program, incorporating an ease of use seldom seen in CAD programs.

A COMPUTER AIDED DESIGN PROGRAM





All of Athena II's commands are well documented, in a large well organized manual. A wealth of commands make any task a breeze.

FEATURES:

Auto Dimensioning.
B-Splines:
256 Layers. (16 Colors)
Circles: (Radius, Diameter, & Three Point).
Arcs: Ellipse, Elliptical Arcs, & Fillets.
Lines: Horizontal, Vertical; Parallel, Perpendicular, Tangental, Line at an Angle, Line between Lines, Rectangles, Line Tangent to Two Circles, Etc.
Transformation: Move, Copy, Rotate, Scale, Mirror, Trim (Adjusts any line or circle so that it will exactly intersect with any given line or circle. No magnifying an area to see if your lines exactly meet!)
Parts: Create "Parts" from an existing drawing, and save them. Load "Parts" from libraries, and place them, anywhere, on any drawing.
Isometrics: Semi-automatically transform existing drawings into an Isometric View.
Plus: Grids, Fills, Erase, Text, Points, Files, Information and full Zooming capabilities.

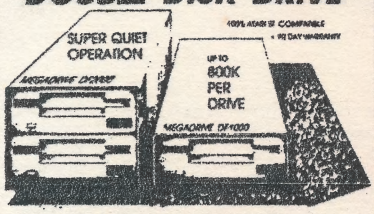
Requirements:
An Atari ST series of Personal Computer, with a minimum of one Meg RAM and TOS in ROM (520ST upgraded to 1 MEG, or a 1040ST) A 3.5" Double sided Disk drive, SF-314 or equal, may be copied to two single sided diskettes for use with a SF-354 or equal.

Atari 520ST, 1040ST, SF-354, and SF-314 are registered trademarks of Atari Inc.

iliad
SOFTWARE, INC.

435 West 920 North
Orem, Utah 84057
Tel: 801-226-3270

DOUBLE SIDED DOUBLE DISK DRIVE



FOR YOUR ATARI ST

C-SYSTEMS — MEGADRIVE

DF 2000 $449 POST PAID
- DOUBLE DISK DRIVE
- DOUBLE SIDED
- UP TO 84 TRACKS
- INTERNAL POWER SUPPLY
- 3-1/4" H x 6-1/4" W x 9-1/2" D

DF 1000 $249 POST PAID
- SINGLE DISK DRIVE
- DOUBLE SIDED
- UP TO 84 TRACKS
- INTERNAL POWER SUPPLY
- 2" H x 4-1/4" W x 9-1/2" D

ALSO FROM C-SYSTEMS

2 MEG MEMORY BOARD — $495 POST PAID
4 MEG MEMORY BOARD — $895

90 DAY WARRANTY
DEALER INQUIRIES INVITED
TO ORDER: POST PAID SEND CERTIFIED CHECK OR MONEY ORDER
(COD ORDERS $04 DEPOSIT AND $5.00 COD FEE)

C-SYSTEMS • Box 333 • 3724 Kentucky Dr. • West Jordan, Utah 84084
Phone 801-969-4336

JEFFERSON SOFTWARE

Modula 2

Development System V1.0

Our compiler is based on Wirth's latest, the 3rd ed. In addition to the compiler, linker, debugger*, and object code decompiler, we have added SOURCE to the libraries to encourage development in Modula 2.

We show examples of how to use Modula 2 to program in GEM, including AES windows.

Learn how easy it is to translate C into Modula.

Benchmarks show our compiler is fast, and you can fine tune your programs to make them even faster and smaller.

$39.95 + $3.00 Shipping

*We are still adding features to V1.0. You still get the best compiler around, with updates available for a SASE and disk, or by sending $10.

Jefferson Software
12416 N. 28 Dr. #18-236
Phoenix, AZ 85029-2434
(602)243-3106 voice
(602)258-7205 Phase Computing BBS

ASTUN

Editor/Publisher

Ralph Porter

Associate Editior

Brad Davis

Advertising Sales
And Circulation

Ralph Porter

General Office,
Subscription Services,
Catalog Orders

6157 South 700 West
Murray, Utah 84123

Subscriptions

US$18.00/12 issues

US$50.00/12 issues w/Disk

Single Issue US $2.50
US$6.00 w/Disk

Canadian Orders add 25%
All Other Countries add 50%

ADVERTISING RATES

Back Cover $40.00
Inside Cover (Either) $30.00
Center Two Page Spread $50.00
Full Page $20.00
Half Page $15.00
Quarter Page $10.00
Business Card $7.00

ASTUN is published monthly as a service to Atari ST user's and to user's groups. Opinions expressed in this publication are those of the author(s) and not necessarily those of ASTUN or it's staff. All articles and programs published are automatically placed in the public domain unless specifically stated otherwise. Public Domain material may be reproduced at will provided ASTUN and the original auThor are credited. Non-Public Domain material must have prior written permission for reproduction.
ASTUN is not affiliated with Atari in any way. Atari and ST are trademarks of Atari Corp. Other trademarks which may appear; GEM &CP/M (DRI), IBM & PCDOS (IBM), Unix (Bell Labs).

TABLE OF CONTENTS

CES Notes 11
by Ralph Porter

Editors Page 4
by Ralph Porter

EZ Tracks Review 9
by Mike Seeley

Gem Treasures in Basic 16
by Gordon Billingsley

IBM Compatibility for your ST . . 13
A Product Announcement

Paradox MS.EM Review 12
by Rick Snevel

RS232 Port Definitions 26
by Al Hyer

ST Register 19
by Jeff Gortatowski

ST to Analog RGB Moniotr 18
From The Astralian Atari Gazette

ST Update 5
by Frank Sommers

Word Perfect For the ST 14
A Product Announcement

EDITORS PAGE

by Ralph Porter

The Past Year and one half of ASTUN's existence has been an interesting one. As Editor and publisher I have learned much about the ST computer and have learned to appreciate it's potential.

At this time with some mixed emotions I am going to anounce to all readers that ASTUN is about to cease publishing, at least in it's present form.

It was my hope that our readership would continue to grow. This has not been the case. In ASTUN's early days it had around seventy five subscribers, today it has around twenty five.

With this small base I can no longer afford to continue this business. ASTUN has been a hardship on my personal and family finances. It has also caused some unwanted feelings between family members. As a result the decision has been made to cease publication.

I had also hoped to get more support from readers, in supplying articles for publication. Some did respond and to them I offer my thanks. I hope you will also continue to support the new publishers of this magazine.

ASTUN has been exchanging newsletters with seventy clubs from around the world. It has also been supplying the magazine to the one hundred members of the ST Users Network. This fine club has agreed to continue publishing this magazine. The next issue will be a joint effort between myself and the STUN group. Following that issue STUN and its members will be the producer. I will also continue to offer support.

Along with this change will be a change of address for corespondence and a name change for the magizine.

The executive board of STUN has decided to drop the leading 'A' from the title of this magizine and to have it fully represent by name the name of their user group.

Paid subscribers may continue to receive this new publication untill the expiration date on their mailing label. Disk subscribers will also receive the STUN disk of the month. Subscribers who find the new publication unsatisfactry may also request a prorated refund of their remaing subscription.

I have been involved with STUN since its orgination and feel that they will represent you well.

Upon expiration of your subscription continued reception of the magazine will require membership in STUN. Look for more details in the next few issues.

Like ASTUN, STUN trully supports the ST computer line. They do not offer support for Atari 8 Bit or any other brand of computer. We feel that as an ST user you will appreciate their efforts.

ST UPDATE

Latest News in the ST World
by Frank Sommers
Reprinted from Current Notes

TRUST AND THE MARKET PLACE

Are the Poles Wrong? -- Limited polls about Atari and Jack Tramiel are producing results almost identical to the polls on the Presidents credibility and popularity, where the latter's word is not trusoted by 65% of the people, and yet 66% of these same people beleive he's the right man for the job. (Admittedly the size of the poll is minuscule and the margin for error large.) But uncertainty about where the leadership of Atari is taking the company and our favorite product mounts by the month. Barely was the bloom off the product announcements at CES before the first vestiges of doubt began to produce blossoms. Disappearance of the $499 version of the Atari PC helped trigger the mud slide. Now the spring is fully here so are the mounting convictions that what Atari says has little relation to what Atari does.

HARDWARE

Laser Printer -- The hardliners among the sceptics say that while the laser printer was admittedly a little wispy when it was announced, with minimal data about how it was to be configured, and no date for appearance, lately the laser printer has been coming out of its cocoon and transforming itself from vaporware to nowhere. No contacts have been agreed upon for the much discussed Atari desktop program. How long does it take to produce a new and innovative desk program? System Three in Canada has been working on one for the ST for 2 years. Even with the top of the batting order from Batteries Included coming over to give it a boost, it is not for this quarter. How long before Atari can have a program to run on their laser printer? Tales from the Tramiel woods would have us beleive that "everything is all of a package", and the Megas will appear, along with the blitter, and right behind if not along side will be the laser printer and over in the other corner with a puff of smoke will appear the PC line.

What are the facts, or the nearest thing to them? The laser printer is unlikly before Xmas! Remember the CD ROM? The market and the price collided in Atari's mind and the project is on a shelf someware wating. The laser faces the same uncertainties. Then, if a one-man survey concludes the desktop market is there and if the software driver materializes, then th ok for the production would be forthcoming. But the production cycle is three months long, and no one has yet confirmed that Jack has ok'd production. The blitter chip, you remember, was declaired ready to go last November, but wasn't ok'd for factory treatment until January. As of this writing, it has not been seen in the stores.

MEGA ST -- Here the news and wispers is better. Dealers are being told that the MEGA ST, the 1 meg upgrade chips for the 520 and 1040 ST's, and the blitter will all arive about the same time, in May. The boats are in the harbor, FCC approval's obtained, and the band is ready to play. Production models of the MEGA ST with blitter have already been shipped to Canada (we have talked to no one who has seen the production model there), along with the Atari PC. How do you sum this? Positively. By June you should be seeing them at dealers around you, minus the printer that was to be the critical part of the desk top package.

IBM Emulators -- A distillation of what's been heard and what's being said suggests that the Atari "blue box" has not been discarded, but with the advent of new emulator products who's to say it won't be. Paradox Software is releasing a beta version 2.0 of its IBM emulator MS.EM. This according to Paradox will cure the bugs in the origional version. Avant Guard Systems, owned by the Teals in Jacksonville Florida will market a software emulator at the end of May that once booted will turn your ST into an IBM PC running at 80% the speed of the ST with 99% compatibility. (Ginny Teal, one half of the partnership with husband William suggested, "99% is correct, but you better say 95% we'd like to be conservitive about this.) They have tested 400 or so pieces of IBM software and found only BASIC A, because it uses special IBM code, will not run on the emulator. All of the "toughies", FLIGHT SIMULATOR, SIDEKICK, MULTIPLAN, DBASEIII, LOTUS 1-2-3, SYMPHANY, ENABLE, DOLLARS AND SENSE, etc., will. IB computers of Portland, Oregon has agreed to make the 5.25 drives for the ST, for those who chose not to port software over to their 3.5 drives. The program called PC-DITTO will cost $89.95; Avant Guard Systems, Pablo Point Dr., Jacksonville, FL 32225.

8-Bit Emulator -- Some of you have seen, briefly an 8-bit emulator up on the BBS's, for a day mabe. Reportedly it was put there to get Atari's attention, who for 3 months or so had declined to answer the author's calls, letters, etc. Finally there was a reply, the day after the BBS appearance. . A copy of the emulator program, for running basic language 8-bit ware on the ST, was sent, and subsequently lost by Atari. Hopefully, negotations continue. Reportedly all the author wants to do is make the program available in the public domain. If this is true, one wonders, does Atari have a case of that perilous computer disease, "If it isn't invented here it doesn't exist"?

PC and Others -- The Atari PC will be shown in quantity at the June CES in Chicago and two weeks later will be on the dealers shelves. The TT (thirty-two bit) is scheduled to be shown at the November Comdex. Niel Harris has stated thet the Enhanced ST with better "resolution than we currently have in monochrome" and the TT capability is still alive and waiting. It will be a stand alone machine. Are these two items one and the same?

ATARI VS SOFTWARE

Do We Mean What We Said?

-- The Denver Rocky Mountain Atari Fest may have been the most sucessful show to date. Only a couple of months old, it brought together hald-a-dozen clubs and exposed them to "the truth" (See Niel Harris report in the issue). The Atari technical rep who participated in a major seminar was as good a spokesman as Atari has fielded to date. He offered, in the new Soviet style of openess, unfettered information on compatibility problems with old software and the blitter chip. Except for golfers who may have trouble seeing the club swing because of the speed of the graphics, nothing of consequence. (CN Golf Team, keep the blitters off your machines if you want to stay on the team --next tournament end of May.) He explained without fear of being fired, why Atari had moved the power switch on the 520St to the posisition it has on the 1040. He admitted that the IBM emulator box from Atari was "still being worked on", acknowledging that it should interface with the cartridge port on the ST? A spokesman of unfettered info. Niel Harris followed him in, as you read elsware in Greg Anderson's remarkable rendention, and outstanding unburdening of "Things You Wanted to Ask But Nver Got Answers To". The result? Attendees said it was hard to improve on. Consequence? Word from the underground is that Atari, possibly on Harris' recommendation, has decided to unburden themselves, and drop out of Atari fairs. Watch for fairs in Dallas and Seattle in May to verify this.

What does Atari say? "Support of user groups is one of our major priorities." What do they do?

IMG Scanner -- The "outer space" device, cited in the last issue, that would scan any picture, document, graphic, etc., and move it into your computer's memory for manipulation, has appeared recently, or its product has, on CompuServe. Several screens have been downloaded by various readers, and the general comment is "I could hardly beleive my eyes, the quality was so remarkable." Soft Logic has acquired the scanner, which in early form was selling by Seymor-Radix for $59.95. Soft Logic will package the product with PUBLISHING PARTNER and possibly service early orders, turned over to them, at the origional price. PP has proven itself to be an outstanding customer-support company, and alliance, therefore is good news to those of us interested in incorporating a scanner into our systems. Latest reports from people who had been using the proto-type version were that the results are best in low resolution. A footnote:Soft Logic will be destributing fonts for the PP from the Font Factory of Phoenix.

Unix, the PX and Fleet St -- Dave Beckmeyer is willing, is writing? Unix 5.3 for the ST. Market? Ma Bell pays circa $10,000 for its terminals. With Unix 5.3 on th ST a rather inexpensive replacement would be at hand. Autumn will see ST's in military PX's here and abroad.

1ST WORD will be marketed by its British parents in th U.S. and Canada and not by Atari. FLEET STREET PUBLISHER is reportedly powerful, in not as practical as PUBLISHING PARTNER, but it will not support a laser printer.

Batteries Included -- EA bought most of BI's fixed assets, inventories and artists contracts. The BI unborn product that has been the most morned and wondered after is PAPER CLIP ELITE. One of its two authors, Dan Moore, has reportedly received a contract offer for EA, which may not be significant enough for him to accept and to continue working on the project. Much of the inordinate delay is said to accrue from problems with GDOS. Though Atari and Len Tramiel maintain that GDOS is "finished", released, and out there and working, PUBLISHING PARTNER's authors, for one, will tell you that ain't neccessarilly so. (They had to design their own version of GDOS in order to bring PP to the market.) PCE could be out "in several weeks" if it chose to forego GDOS and be Epson only compatible. PCE has been billed as "the best"; to be that it requires GDOS to achieve its outer limits. If you load the current beta version of PCE with the current GDOS, the latter gobbles up 500K, leaving PCE naced ont the beach before the first page is typed. Once again is, "What Atari says about a new product, have any beating on what it is when it finally appears?"

THE BOTTOM LINE

So why the antagonism out there to what Atari is doing for us and the country? It may be caused, partially, by fear. Fear that the machine, old and dew owners find so easy to dedicate themselves to, that somehow this machine may not survive. A glance at 8-bit owners and both the syndrome and the possibility are credible. Thus we, who are accurssed of being the most sophisticated hackers in the computer trade, and the deadliest, to software companies, of pirates, are equally alert to what we consider to be missteps in the directing of our computer's future. Decisions like the one that "emulators have only minimal markets" from Jack and Co. antagonizes us. Yet, after we have seen and used the emulators we grow boared. Minimal market?

So while you don't like being misled, and you know that while politicians can get away with it, CEO's and their companies don't over the long haul, later on we are pleased, if we conclude we have been well led. While the jury may or may not still be out on the President, it clearly is on Jack Tramiel and Atari.

STUN BBS

(801) 964-9764

1200/2400 BAUD

EZ TRACKS

FROM HYBRID ARTS:REAL-TIME MIDI SEQUENCER FOR THE ST
by Mike Seeley

A 20-track real-time only program aptly named "EZ Tracks". While not quite up to the wowie-amazing category of Hybrid's "DX Droid" for Yamaha synthesizers, the price of this product ($65 retail) will undoubtedly pour some money into the hands of Hybrid Arts - and deservedly so. While other software companies have belatedly been joining the ST bandwagon, Hybrid has supported the ST since Day One, and I hope to see them reap some profits from it.

"EZ Tracks" is a basic sequencer that will run on either the 520 or 1040ST - you must have the TOS in ROM to run this one. You have twice the note capacity if you own the 1040 or a 520 with one megabyte added. Note capacity is around 60,000 - enough for about a whole album! The program runs on mono or color systems (in medium resolution), and is set up similiar to a multi-track recorder with play, pause and stop controls. In fact, there's a whimsical little recorder drawing that sits in the bottom right-hand corner of your screen (t make computer-phobe's feel at home I guess). Oddly, there is no record button - to record, you select which track you want to record on with the mouse and push play. The program automatically starts recording on that track unless you have protected it. When done playing your part you may click on the "keep" button to save the track, or if you thought your playing bit the weenie you can simply push play and start again. I managed to "wipe" a few tracks until I learned to protect them - oh well, that'll teach me not to read the manual first. There is no fast forward or rewind controls, but by clicking the mouse on the "tape counter" you can move the sequence up or back to where you want it.

I should add here that "EZ Tracks" makes full use of GEM pull-down menu's and desktop features, and of course the mouse. In fact, it would probably be impossible to use this program without the mouse. A real nice feature is the ability to keep those drop- down menu's from dropping down. By selecting this function the menu's will only drop down when you click on them - presumably this will keep you from "hitting" them every time you go to Track 1 at the top of the screen. Mac owners and beginners will love this feature although I found it confusing at first. Since it's selectable you can have it either way.

There is a limited editing menu on "EZ Tracks". Basically, you can autocorrect your playing (quantize it) to the nearest 1/4-note, 1/8th-note, 1/16th-note or 1/32nd note. You can't edit out other data such as pitch bend or

program changes, but you could put these on a different track altogether (after all there are 20!). Mercifully, the auto-correct is non-destructive: the auto-corrected version of your track is "bounced" to another track.

Speaking of bouncing, yes, "EZ Track" does this too. You can take any two tracks and merge them together into another track, thus freeing up those tracks for more stuff. You can't un-merge the tracks after doing this, so if you're unsure save it to disk first. You can copy tracks too. There is no step-mode data entry in "EZ Track", but considering the auto-correct ability most folks won't miss it. You can also slow down the tempo for recording and speed it up for playback to improve your playing. The tiny little metronome clicking away on the upper right of the screen is a neat graphic touch - and you can set the resolution within 3/10 of a beat per minute - better than any drum machine I've ever seen. You can also transpose the pitch of the entire song up or down, and a "gas gauge" keeps track of how much memory you've used (this is an ST - don't worry about it!).

Song files are saved as ".sng", and a few are included with the program for demo purposes. They load on quickly, although I haven't had a chance to see how long a very large song would take.

Each MIDI track can be set to any of the 16 MIDI channels or to MIDI Omni mode so that it receives all MIDI channels, and each channel can be muted individually. Another nice feature is the built-in metronome (from your monitor's speaker of course) which can be muted. "EZ Tracks" will work with any MIDI keyboard and recognizes all the standard MIDI commands. Any MIDI drum machine should have no problem syncing up to the MIDI clocks outputted over the MIDI line, however since I don't have a MIDI drum machine I can't say how "tight" they sync up. Those of you who have non-MIDI drum machines will need an extenal hardware adapter such as the Korg KMS-30 synchronizer to convert MIDI clocks into the info your machine needs. "EZ Tracks" cannot be controlled by an external source, such as another computer, drum machine or tape sync, but hey! what do you want for $65!

Summing up, "EZ Track" is an excellent buy for the money. While I don't think every recording studio in the world will rush to buy it, the thousands of ST owners who are also amateur musicians will jump on this product. The pro musicians might want to keep in mind that they can upgrade "EZ Track" to Hybrid's "Miditrack ST" (the 60 track pro version) when it's released sometime soon (?!).

ASTUN
BBS

ONLINE 24 HOURS
EVERYDAY

(801) 266-0748
1200 BAUD

CES NOTES Summer 1987

The 1987 Summer CES is now behind us. Following are a few notes taken from various reports on the show.

Atari entered the show with the theme "Flying High With Atari". To emphasize this they topped their both with a real Cessna Airplane. In it's cockpit was a large screen Monitor displaying Flight Simulator II. The story behind the show for atari was Games.

They did not show the new Mega ST, Laser Printer, Blitter Chip or 1200 Baud modem. They did have the PC there but the emphasis was on the game machines.

Press release headlines were as follows. "Atari Announces Aggressive New Marketing Campaign to Support Video Games and Personal Computers", "Atari Announces 40 New Games and Licensing Agreements with Computer Game and Arcade Companies", and "Atari to Begin Shipment of New XE Game System Bundled with Three Popular Games".

The advertising campaign will include comic book ad's, Late year TV Spots and Print Ad. Only in the last quarter will the ST computers be promoted.

Else ware in the show the following products were being displayed for the ST computer.

SCAD - A 2-D drafting program that has 16 memory pages. This program by Xetec will sell for $99.95.

MasterCAD 3-D by a group from Venezuela. This $199.95 program will convert any 2-D image into a 3-D one. It requires 1-Meg and a monochrome monitor.

Master Tracks Pro by Passport Designs is a 64-trac midi sequencer that has received rave reviews on the Macintosh.

Computereyes showed a 16 grey scale video digitizer. This product will sell for $149.95. It will compliment the color version introduced last year for $249.95.

Print Master Plus now has a new set of fonts and boarders labeled as Game Gallery. No price was given.

Mindscape showed two $29.95 games. Q-Ball, a 3-D Billiards game and Plutos, A shootout using rockets to destroy an enemy mother ship.

Barataccas from Psygonsis of England is a graphic adventure type game.

Electronic arts was showing an enhanced version of their Music Construction Set. They will also distribute First Byte talking Software. This includes a version of Mad Libs and a math educational package.

Logical Design Works had a series of Gambling Programs.

Canada's Jagware showed Allien Fires, a 3-Dcolor only space maze.

The people at Accolode had a driving simulator. It features five cars. A Porshe 911, Ferrari 308GTS, Lamorghini Countach, Chevrolet Corvette, and Lotus Turbo Esprit.

Avatex showed its $319.00 2400 Baud Modem.

Also available was fancy covers and a credit card size calculator featuring the fuji symbol and Atari Name.

It was interesting to note that ST's used in demonstrations out numbered Amigas by about six to one. Neither Apple or Commodore had booths at the show.

Atari used the show opening to announce a 2 for one stock split plus a first quarter sales of $65.1 Million.

Paradox Introduces MS.EM

A review by Rick Snevel
Reprinted from TAP Newsletter from the Atari Peekers

So, you want IBM compatibility for your Atari St computer? Well, if you thought that software-based IMB emulator MS>EM (stands for MS-DOS Emulator) from Paradox would be your answer-FORGET IT! This product looks as if it was released too early (probably to bead Atari into the world of IBM emulation on the St.)

I had very high hopes for this product, and was ready to buy it the day it came to B&G. Boy was I in for an eye opening experience. The first MS DOS that we tried to load would not work. Finally, after hours of trying different things and different DOS's we got one that worked. Then the fun came. None of the software we have for it right now (which is a lot since I know someone with both an IBM and an ST) would work. I tried Word Star, Sidekick (which only works a little the first time you try to use it. Even the instructions say you shouldn't trust the calculations it does after the first time, GW-Basic, Lotus 1-2-3 (To be fair, the Docs. say that Lotus won't work), and many others.

I would have bought if it would have run a BASIC (it does supposedly run a BASIC called APC BASIC, but nobody I talked to had ever heard of it.) The next frustrating obstacle is it's speed, or lack of it. This is so slow that it seams as if you are working on a

2-bit computer(?), but not a 16-bit. It does seem to be faster in Monochrome mode than color though.

On the positive side Paradox does seem to be willing to help out the user with any problems that should arise. We called enough times that they gave us there toll-free number which was a big savings on the phone bill. The people there (and you can usually talk to the author of the program) are very friendly and try there best to help you. But as of this writing (Feb. 87), I can not recommend this program for anyone at this stage. It is totally useless.

By the way, the version reviewed here was 1.1, not the original 1.0. The fix is available on the T.A.P. BBS for anyone with the MS.EM program already. One final thought, when running this fix, make sure you do everything exactly correct, because the disk is copy-protected. We tried our best to back this program up, (I don't like doing something like this to a disk that is not backed up somewhere), but whenever we ran the copy, it just prints, "copy, copy, etc." down the entire screen. So, until this program is fixed to run more software, it would be a waste of money to buy.

EDITORS NOTE: Version II of this Program Should Be Available By the time you read this. Several Bugs are said to be fixed in the new release.

IBM COMPATABILITY

IBM Compatibility for the Atari ST

Avant-Garde's IBM PC emulator for the Atari 520 and 1040 ST is now available for purchase. pc-dito imitates an IBM PC with monochrome, color, serial ports and parallel ports (or an IBM PC XT with optional Atari hard disk). A sample of some of the most popular IBM PC products which run are:

Enable	Word Perfect
Framework	Microsoft Word
Lotus 1-2-3	Flight Simulator
Art Studio Draw!	PC-Stock
EasyCAD	Timeline
Superkey	DesqView
IBM DOS (all)	GEM
Managing Your Money	Symphony
Chase Spectrum	Finance Manager
Dollars & Sense	Sideways
Multiplan	WindowDOS
PC Window	GW Basic
Paint Works	Swift Tax
Sidekick	dBase (all)

The suggested retail price is $89.95. This price includes a free update and full support from Avant-Garde System.

WordPerfect for

WordPerfect Corporation introduces WordPerfect for the Atari ST.

The following is a brief list of features for this professional word processor:

Compatibility

File compatible with WordPerfect 4.1 for the IBM PC and other computers, allowing for direct document transfer to and from the ST without losing document format. Function keys are defined the same between versions, for increased ease of learning.

Footnotes/Endnotes

Footnotes and endnotes are automatically numbered and renumbered as you edit. Footnotes are properly placed at the bottom of the page, and endnotes are compiled at the end of the document. There is no limit to length, as all notes can overflow the current page if you designate.

GEM Interface

WordPerfect fully supports the GEM interface. Virtually all functions may be easily accessed with either the mouse or the keyboard. Desk accessories are fully accessible from inside WordPerfect.

List Files

A complete set of disk utilities is included, for total file maintanance.

Macros

Record any series of keystrokes or mouse actions and recall them with a single keystroke. Macros can be chained or conditional, as well as delayed.

Math

Math mode allows creation of numeric tables in your document, with automatic calculation of subtotals, totals, grand totals, or your own custom math functions.

Merge

Merge can be used to automate many office proceedures, including forms, labels, contracts, and other time-consuming tasks. The Merge feature may also be combined with macros to create powerful user-defined functions.

Paragraph/Outline Numbering

Paragraphs can be automatically numbered in several different styles. Or, use the outline feature to simply organize your documents.

Printer Support

WordPerfect supports over 200 printers, including most laser printers. Documents can be printed using true proportional spacing, font downloading, or virtually anything else your printer is capable of.

Speller

A fast 115,000-word dictionary with phonetic and word-template look-up is included. Fully expandable, with legal and medical terms already included.

Table of Contents/Index Generation

Create a table of contents or index for your document, consisting of up to five levels.

Text Columns

Up to five newspaper-style or static text columns may be displayed and edited on-screen.

Thesaurus

Synonyms and antonyms may be displayed for up to three different words at the same time.

Undelete

The last three deletions or series of deletions can be restored at will, at any location you choose.

Virtual Memory

Data can flow onto disk when computer memory is full. No longer are your documents limited by available memory, but only by disk size.

A complete manual

including graduated lessons, a thorough reference section, and a color-coded keyboard template, provides of operation for both new and experienced WordPerfect users.

WordPerfect is scheduled for release this summer. Watch for world-class word processing at your local Atari dealer.

Questions can be directed to Jeff Wilson, Manager of Atari Development for WordPerfect Corporation, at CIS: 72447,3427.
Or write:

WordPerfect Corp.
288 West Center
Orem, UT 84057

GEM TREASURES IN BASIC: polymarkers

by Gordon Billingsley

Polymarkers are pre-defined graphics shapes that GEM allows you to print to the screen by giving only one coordinate pair. You don't have to establish the corners or boundaries of the shape to place a relatively complex shape on the screen. GEM does all that automatically.

For BASIC programmers that makes a handy way to use relatively complex shapes in game programming. Such shapes as squares or diamonds can be printed to the screen in any of four sizes with no more than two values -- and sometimes only one -- to change to move it around the screen or place it in user-selected locations.

Polymarkers also make handy marks for menu items to be selected by the mouse. Future columns will cover using the mouse. I have used polymarkers as building blocks for more complex graphic items. For instance, I used square polymarkers in conjunction with periods to construct graphic dice for a game of chance.

The program lines for polymarker routines will be among the longest we'll cover in this series. That's because we must have routines to change the shape, the size and the location of the polymarkers.

The routines are set up so that you can select a shape by simply calling a subroutine by the name of the shape. An example would be GOSUB SQUARE to get a square polymarker. The same holds true for the size of the markers. The only time you must literally assign a value to a variable is when you are providing coordinates for the marker to be printed to the screen.

Here are the program lines:

```
10 DOT:mtype=1:goto markertype
20 PLUS:mtype=2:goto markertype
30 ASTERISK:mtype=3:goto marketype
40 SQUARE:mtype=4:goto markertype
50 CROSSBUCK:mtype=5:goto markertype
60 DIAMOMD:mtype=6
70 MARKERTYPE:poke contrl,18:poke contrl+2,0:poke contrl+6,1
80 poke intin,mtype:VDISYS(0):RETURN
90 SMARK:sizemark=12:goto markersize
100 RMARK:sizemark=24:goto markersize
110 LMARK:sizemark=36:goto markersize
120 XLMARK:sizemark=48
130 MARKERSIZE:poke contrl,19:poke contrl+2,1:poke control+6,0
140 poke ptsin,0:poke ptsin+2,sizemark:VDISYS(0)
150 MARKER:poke contrl,7:poke contrl+2,1:poke ptsin,markx
160 poke ptsin+2,marky-int(sizemark/4):VDISYS(0):RETURN
```

Descriptions of the six types of markers:

GOSUB DOT calls up a one pixel dot (the only polymarker for which the size cannot change).
GOSUB PLUS makes a plus sign (+).
GOSUB ASTERISK makes a standard asterisk shape (*).
GOSUB SQUARE makes a square.
GOSUB CROSSBUCK makes a diagonal cross shape like the crossbuck shape on railroad crossing signs.
GOSUB DIAMOND makes a diamond shape with a horizontal orientation. After you call one of these marker shapes you must specify a marker size. The available sizes are as follows:
GOSUB SMARK makes the marker one-half the size of a character on the screen.
GOSUB RMARK makes a marker the same size as a default character on the screen.
GOSUB LMARK makes a marker one and one-half the size of a default character.
GOSUB XLMARK makes a marker two times the size of a default character.

To make the size routines easier to remember for programming, I've named them after clothes sizes: S(mall)mark, R(egular)mark, L(arge)mark and X(tra)L(arge)mark.

The size routines also automatically place marker on the screen, so you must specify a coordinate point for the marker before you call one of the size routines. The variables for that are "markx" and "marky".

Here's an example of creating a double-sized (tall as two lines of characters) polymarker shaped like a diamond:

```
10 markx=40:marky=120:GOSUB DIAMOND:GOSUB XLMARK
```

Note the order in which the various parts of the routine are called or referenced. First: the coordinates are set as pixel values on the screen with variables markx and marky. Second: the shape of the marker is called. Third: the size of the marker is called and it is automatically placed on the screen in the specified location. It's important that these steps be followed in order. If you were, for instance, to call the size first your system would possibly crash because not enough values would be assigned to variables in the GEM routines.

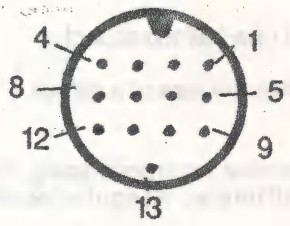
Once the attributes of the polymarker are set they remain the same until specifically changed later. I mentioned the possibility of crashing the system. It is important all-around to be very careful when typing in the program lines provided in this series of columns. GEM VDI and AES routines are extremely sensitive. I honestly think they like crashing--the demolition derby nuts of the Atari world.

In the next column we'll develop some routines for reading the mouse and look at a couple of algorithms for using those routines.

Add an Analog RGB Monitor to Your ST Computer

reprinted from the Australian Atari Gazette

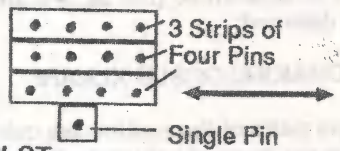
The following Diagram Illistrates how you can have both a Monochrome Monitor and an Analog RGB Color Monitor attached to your ST Computer



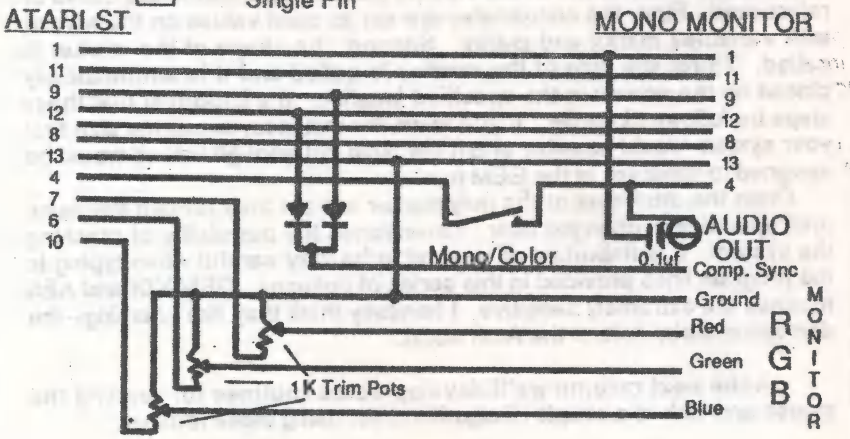
Connector Viewed
From rear of Computer

Pin No.

1 - Audio Out
2 - IF TV Composite Out
4 - Mono Detect
6 - Green Out
7 - Red Out
8 - Ground
9 - Horizontal Sync
10 - Blue Out
11 - Mono Composite Out
12 - Vertical Sync
13 - Ground



Glue Strips Of Header Pins Together Using Super Glue



ST Register

by Jeff Gortatowsky

reprinted from The Acorn Kernel

I know, real ST programming is what you want. However the fact is that there are basic concepts that don't seem to be understood by many people starting out in C. You must be able to read the language and understand the implications of it's constructs, if you want to profit from example code.

With that said, I'd like to discuss the scope of variables. The what? The scope. The scope of a variable determines when the variable may be used in a program statement. Further the scope of a variable determines the extent to which the variable is 'visible' or accessible during the course of the programs execution. This all depends on another buzz phrase, known mostly among compiler writers, as a variables "storage class". Or 'how and where' is the current value of a variable stored and retrieved.

In BASIC on the ATARI 800 and it's kin, the scope of a variable was easy. Any variable once used or declared was accessible to any other part of the program and stuck around till such time as the program terminated. Thus you would be correct in saying the variables in BASIC were 'global' in scope. The entire program could reach and modify and variable. So you might consider BASIC as a small world. And in that world you were master and had 'global' reach when it came to changing variables.

Not so in C. Or at least partially not so. C indeed does have global variables. They have a 'storage class' of global as well. Generally the 'how' and 'where' of the global storage class is to gather all such variables into one chunk of memory that is always accessible. Globals tend to be used a lot by people who don't understand the concepts of C, or the lazy programmer. Lots of globals tend to take up lots of space, confuse the overall flow of the program and create debugging nightmares for any multiple file program. To see for yourself an example of abuse of globals (and other atrocities using the C preprocessor) gander at a few of the programs written in C in any issue of ANTIC.

Any part of your C program can modify a global variable at any stage of it's execution. Further your program may consist of multiple C files all compiled separately and linked together at the end, these global variables are accessible even to the other modules. The C 'extern' keyword is used to declare a name for a global variable whose actual declaration (the one that allots memory space) is in a different file. this lets the code in the current file 'know' or 'see' the real variable. Eventually the linker will see to connecting all these 'extern' variables to the actual memory locations.

So where does C differ with BASIC? In three other 'scopes' you may give to variables. Their names are: "Local", "Static Global", and "Static Local". In contrast to global variables each of these three may modify a variable. Indeed each one defines a limit as to the portion of your program that even realizes a certain variable exists!

Each of the three above mentioned scopes of a variable has it's advantages and it's drawbacks. We'll tackle each one in a moment. First

however we should wonder why we would want to limit the accessibility of a variable in the first place? I'm sure many of you can pick this one up right away. Foe debugging purposes. If we know a certain variable is not able to be reached by a portion of our program, then that portion of code can be ruled out as the source of a bug. That's of course assuming the bug involves the variable in question. Another reason would be the reuse of a variable's name. If you like using the variable name 'loop_counter' for a counter in program loops, hiding each instance of that name from other portions of the program will allow that. Since any one instance of 'loop_counter' may be 'hidden' from another, there is no conflict. Each instance is unique. I'll explain why they may be unique in a moment.

We'll start with perhaps the least understood but most often used storage class for C variables, the LOCAL storage class. Now when I say 'storage class' I'm referring to 'where and how' the variable will be given a place in the computer's memory to reside. Depending on this where and how we also also defining if anyplace else in our program knows of it's existence or it's scope.

Local variables, also know as automatic variables, are declared INSIDE of the body of a function. More formally they are declared INSIDE a 'block' of code. The synonym 'automatic' comes from the fact that the storage for these variables is automatically created and automatically destroyed each time a block is enter or exited. A block is started with a curly brace. When a variable is declared inside a block of code it only functions (exists) as a variable until the end of that block. No further. Once the end of the block is reached all the variables that were declared in that block cease to exist. That's why they are known as 'locals. They only exist in a very localized area of code. Here's an example.

```
my_function()   /* Declare a function start */
{               /* begins a block of code (a new function as well) */
    int a;      /* The integar a is declared inside the block. ie LOCAL */
    char b;     /* Same here for b */
    long c,d:   /* And again for c and d */
    c = 0;      /* code, etc */
}               /* End of block (and function in this case):a,b,c,d vanish */
my_function2() /* Start another function */
{               /* begins a block of code */
    int a;      /* These variables are local to 'my_function2'.  They */
    char b;     /* ONLY exist inside this 'block' therefore are separate */
    long c,d;   /* from the variables in the above function even though */
                /* the names are the same */
    a = 1;
    if(a == 0)
    {
        int a;  /* This variable SHOULD BE completely different from the */
                /* 'a' only declared a few lines ago.  Why?  Because the */
                /* 'if' statement opened a new block of code. ie. "{"  */
        a = 2;
    }           /* End this block, second a vanishes */
}               /* End of bolck 1st 'a' and b,c,d vanish */
```

If you read the comments carefully you should see that these two functions declare a total of 3 integer variables named 'a'. Yet all 3 are unique and do not conflict with each other. Why? because all are found in DIFFERENT blocks of C code. The second declaration of 'a' in 'my_function2' shows this. Now in reality most programers do not choose to declare variables inside of 'if', 'while', 'for', 'do', and 'switch' blocks. But it is valid to do so. The compiler SHOULD NOT complain. Your compiler may not. Many do. Most programmers declare all their local variables right after the opening block ('{') of the function.

So where do these local variables get stored? And how do they vanish and release what memory they took up? Well they get stored on the CPU's stack. What? On the stack Yup. And no, they don't mess up the return addresses of interrupt handling. Here's how it works. NOVICES NOTE: I'm not going to explain the CPU's stack, this discussion assumes you understand what it is and how stacks work.

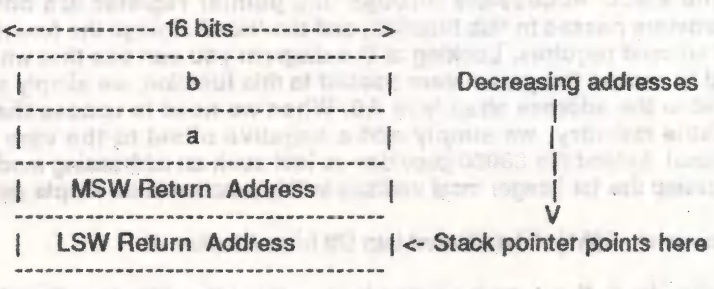
In 6502 assembler you'll recall that when you jumped to a subroutine the system pushed the return address on the stack. The 68000 does the same. Further you'll recall that BASIC has a method of passing parameters on the stack using the USR BASIC keyword. Well 68000 supports parameter passing just as will. When you C code calls a function the parameters specified get pushed on the stack in right to left order, the the JSR or BSR (Branch relative Subroutine) instruction is executed. So the execution of the following C code:

```
my_function(a,b)
```

translates to the 68000 code:

```
move b,-(sp)   * Push the contents of 'b' onto the stack
move a,-(sp) * Push the contents of 'a' (remember right to left)
jsr  my-function  * Hop on off to my_function'
```

so the CPU's stack upon entering 'my_function looks like:

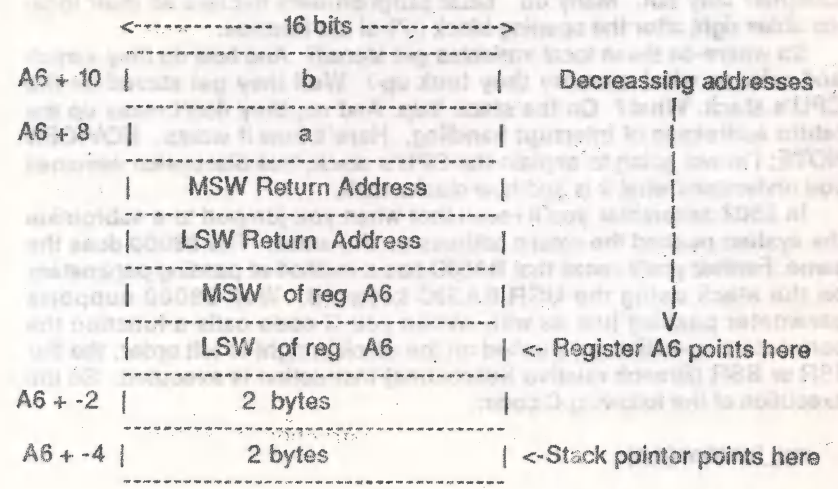


Ok, So now we've reached 'my_function' and the parameters we want that function to deal with are situated just above the return address on the stack. Now we need to make some room on the stack for our LOCAL variables. Remember we said thats where they get stored. The 68000 has an instruction. It's called the 'LINK' instruction. Here's how it works. Lets

say that 'my_function' needs local storage for two 16 bit (2 byte) integer local variables. This is a total of four bytes of storage. The instruction looks like this.

```
LINK -4,A^
```

and the stack looks like this:



Wow! A whole lot has changed in one instruction! What happened. First and foremost there are 4 bytes of 'empty' storage at, and above where the stack pointer is pointing to. If you have not guessed, that's our 4 bytes for out local variable storage. But what's with register A6? Why did it get pushed back on the stack? The reason is that with A6 safely preserved on the stack we can use A6 as a 'pointer' to all the various variables we have on the stack. Accessible through this pointer register are both the parameters passed to this function, and the 'local' storage the function we just entered requires. Looking at the diagram you can see that when we need to access the parameters passed to this function, we simply add an offset to the address already in A6. When we need to access the local variable memory, we simply add a negative offset to the vary same address! Indeed the 68000 provides us just such an addressing mode. So accessing the 1st integer local variable in this function is as simple as:

```
move d),-2(A^)  * Load word into D0 from A6 plus -2
```

Get the idea? If not, go back and look again at the diagram. Keeping the locals on the stack is not dangerous as any change caused by interrupts or other function calls will decrement the stack pointer. The only way to increment it is by exiting this function using the 68000 UNLK (unlink) instruction which undoes all this, restoring (poping) A6, and finally doing a RTS to return.

Why such a detailed explanation? Two reasons. First I believe that even if you never program at the machine level, you should understand what goes on there. The other is less subjective. I wanted you to understand how local variables disappear upon exit from a function. And I hope you can see that above. Whether you can or not, remember that local variables are temporary variables that are valid only within the block of C code they are declared in. Further local variables can NEVER be assumed to hold any known value upon entry to your function. They must be initialized before use each and every time the function is entered.

I said we'd cover the advantages and drawbacks to each storage class, so I should mention the 'hassels' that go with local variables. the most glaring drawback of local variables is also their greatest strength... they are... well... LOCAL! I can't change their value from some other function. But of course C being the 'free wheelin' language it is, lets you do almost anything with a bit of work... and risk. C provides you with global variables if you REALLY need to access a variable time and time again from many different functions. More often than not however only two or three functions really need to access data from another function. this can easily be done by passing the required value of the variable to the function that needs it.

Normally you'd pass a 'copy' of the value and leave the original intact. But locals in one function can be modified directly by another function by passing a pointer to the local variable instead of a copy of it's value. GEM does this A LOT! By using the 'address of' operator '&', you create a pointer to the variable. Passing this pointer to GEM allows GEM to store such things as mouse coordinates in your function's local variables. It's not an ideal situation. Allowing a different function to 'twiddle' with your local storage can be dangerous. But when a function needs to return more that one result there is little choice but to use this method of resort to globals.

GEM programming does require some globals. It's an unfortunate situation caused mainly by the brain damaged 8088/IBM-PC under-pinnings that ST-GEM was based on. However code you create need not conform to this. Keeping your use of globals to minimum will force you to think out your program better and help decrease you chances of sloppy coding.

'Static locals' are mearly local variables that don't go away! Thats right. They retain their value each time the function exits. To declare a static local you simply precede a local declaration with the 'static keyword. For example:

```
my_function()
{
    static long score;
  /* Your code goes here */
}
```

The long variable 'score' in the above function is only known to exist by the code within that function. However the variable, and hence it's value never disappear like normal local variables. When the function

'my_function' is entered the value of 'score' will be the same as the last time the function exited. If this is the first time the function is entered then the value of 'score' will be zero. Your C compiler will (should) take care of that. If you wish any initial value other than zero, you must explicitly initialize the variable when you declare it.

As you can see, this can come in handy for values that need to be retained from one invocation of a function to the next yet still only be 'known' by that function.

Disadvantages? Well, static locals always take up memory space. Further accessing static locals can be slower than addressing the equivalent auto local variable. It depends on your compiler. Also static locals can't be held in registers, so no 'register static' storage class is available for those that need a bit more speed.

OK. What about 'static global's? Static globals come into play if your program consists of multiple source files. Static globals are only visible in the file in which they are declared. All the functions written within that file have access to this variable. However the variable is 'hidden' from any other files, and therefore functions, that reside in the other files that make up the program. Not even using the 'extern' keyword in the other file will allow them to 'see' or access this variable. An example:

```
        FILE #1
-------------------------------------------------
long glb;              /* Global to entire program */
static long st_glbl;   /* Global to this file ONLY */
main()
{
    glbl = 20;
    st_glbl = 400;   /* Both accessible here */
}

a_function()
{
    glbl = 1;        /* glbl AND st_glbl are still accessible */
    st_glbl = 4;     /* because this is file #1 */
}
-------------------------------------------------
      END FILE 1
=================================================
        FILE #2
-------------------------------------------------
extern long glbl;    /* this allows function in file 2 to 'see' glbl */
                     /* 'st_glbl' is only accessible by code in file 1 */
                     /* so we can't 'see' or use it in code here */
my_func()
{
    glbl = 2;   /* We can still access glbl, but not st_glbl */
}
-------------------------------------------------
      END FILE 2
```

Lets wrap this up for now. As we hope you've seen, C provides for some very flexible as well as safe storage for variables. It is important for you to layout your program carefully so you can take advantage of each storage class's unique characteristics. Don't abuse globals. The ease of programming 'quick and dirty' with globals will only lead to headaches as your program grows. By correctly choosing the scope of each variable, you will help decrease debugging time, and maximize your memory usage. Small code changes become easier, and code will become much easier for you and others to read.

The Following Taken from The Australian Atari Gazette.

MIKSCH'S LAW:

If a string has one end, then it has another end.

WYSZKOWSKI'S THEOREM:

Regardless of the units used by either the supplier or the customer, the manufacturer shall use his own arbitrary units convertible to those of either the supplier or the customer only by means of weird and unnatural conversion factors.

THE SNAFU EQUATIONS:

1. Given and problem containing 'n' equations, there will always be 'n+1' unknowns.
2. An object or bit of information most needed will be the least available.
3. Once you have exhausted all possibilities and fail, there will be one solution, simple and obvious, highly visible to everyone else.
4. Baldness comes in waves.

SKINNER'S CONSTANT (FLANNAGAN'S FINAGLING FACTOR):

That quantity which, when multiplied by, divided by, added to, or subtracted from the answer you get, gives you the answer you should have gotten.

Some information on RS232 comunications cableing.
by Al Hyer

Following is a list of EIA pin numbers, their names, and functions:

pin	name	><	function
1	FG		frame ground
2	TD	>	transmitted data
3	RD	<	received data
4	RTS	>	request to send
5	CTS	<	clear to send
6	DSR	<	data set ready
7	SG		signal ground
8	DCD	<	data carrier detect
9		<	positive dc test voltage
10		<	negative dc test voltage
11	QM	<	equalizer mode
12	SDCD	<	sec. data carrier detect
13	SCTS	<	sec. clear to send
14	STD	>	sec. transmitted data
15	TC	<	transmitter clock
16	SRD	<	sec. received data
17	RC	<	receiver clock
18	DCR	<	divided clock, receiver
19	SRTS	>	sec. request to send
20	DTR	>	data terminal ready
21	SQ	<	signal quality detect
22	RI	<	ring indicator
23		>	data rate selector
24	TC	>	ext. transmitter clock
25		>	busy

> = to modem, or DCE
< = to terminal, or DTE

The following diagram should help to understand DTE and DCE:

DTE DCE DCE DTE

terminal--------modem-----modem-------computer
>pin 2 TD-------pin 2 RD----pin 2 TD-----pin 2 RD
<pin 3 RD-------pin 3 TD----pin 3 RD-----pin 3 TD

DTE = data terminal equipment
DCE = data communications equipment

The above signals range from + or - 5 volts to + or - 30 volts

Following are some data terms and deffinitons:

Simplex

One way transmission

like a slaved receive only printer

Half Duplex

Can send and receive but only one way at a time

The host sends data to terminal but has to stop sending when terminal is typed on

Full Duplex

send and receive at same time.

You can type in to the host while it is outputing data to the terminal and the host will still receive what you typed. Data can flow both ways simultaneously

Synchronous

Data transmission that uses clock signals to keep both ends in syncronization

Data signals are not bracketed with stop and start bits

Asynchronous

Data transmission does not use clock signals to keep both ends, the sender and receiver, in synconization

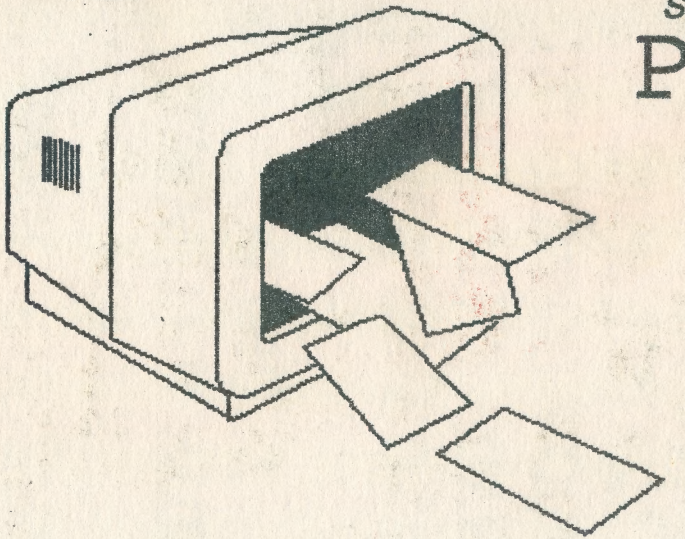
The data will have a start bit before it and from one to two stop bits after it

POPE'S
COMPUTER STORE
967-1037

SOFTWARE 20% OFF
COMPUTER COVERS
1200/2400 MODEMS
DISK DRIVES
STORAGE CASES
DISKETS
PRINTMASTER +
GR LIBRARIES (6 TOTAL)
$5.00 EACH OR
3 FOR $12

THE SOFTWARE HUT
COMPUTER CENTERS

111 East 300 South 355-0066
1839 West 3500 South 974-0110



SoftLogik's

Publishing Partner

On special this month. Only:

$99.00

Atari 1040ST
Color System $995.00
Mono System $795.00

Atari 520ST
Color System $795.00
Mono System $575.00